













Visievorming op StUF-onderlaag

Henri Korver
StUF Expertgroep 16 maart 2016
La Vie, Utrecht


Bestaande RFC's

Issue-id		Title
RFC0112		Definieer een speciaal element <isEen> tbv subtypering
RFC0123		Stuurgegevens stuf:functie van 30 naar 80 posities
RFC0126		Optioneel verklaren van gebruik attribuut stuf:functie in vrije berichten ...
RFC0145		Verwijderen attributes aantalVoorkomens en aardAantal
RFC0152		Ontbrekende foutcode in de StUF standaard
RFC0155		Toevoegen 'StUF:functie' attribute aan attributegroup 'StUF:entiteit'
RFC0324		Samenvoegen Bv03 en Bv04 in één berichtsoort
RFC0333		Versimpelen synchronisatiebericht
RFC0345		Patchnummer in een bericht
RFC0391		Waardenbereik XML-attribuut “entiteittype” uitbreiden met namespace qualifier.

RFC's n.a.v. SIG-rapport

Issue-id		Title
RFC0404		Gebeurtenis opnemen in kennisgeving parameters
RFC0405		Maak samengestelde kennisgeving deprecated
RFC0406		Verwijderen waarde "zaakInfo" in vrije berichten
RFC0413		Overbodig maken nillable="true" in StUF-schema's
RFC0415		Verbieden van het gebruik van attributes in StUF-berichten
RFC0416		Vervang 'fixed'-waarde in attributes door enumeration van één waarde
RFC0418		Introduceren van wildcards in StUF-bevragingen
RFC0419		Incorporeren van andere formaten (e.g. JSON) naast XML
RFC0134		Stuurgegevens optioneel maken

RFC's n.a.v. GAB

Issue-id	Title
RFC0121	Gelijktrekken datums met stelseldefinitie
RFC0409	 Toevoegen "waardeBestaat" aan waardenbereik van noValue

Nog in te dienen RFC's

Issue-id	Title
RFC0XXX	Faciliteren van REST-interfaces
RFC0XXX	Berichtsoorten optioneel maken: <ul style="list-style-type: none"> • alleen content-model is nog verplicht
RFC0XXX	Contentmodel optioneel maken: <ul style="list-style-type: none"> • alleen elementen die corresponderen met attribuutsoorten zijn nog verplicht
RFC0XXX	Aanbevelingen uit Haagse Pilot



StUF-familie

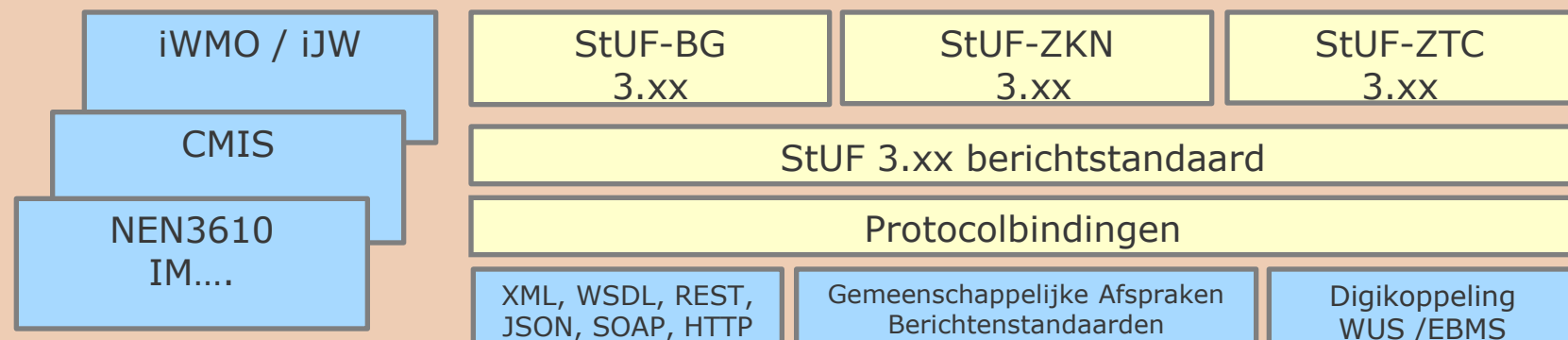
Externe aansluitingen / ketens

Handels Register	Mijn Overheid Lopende Zaken	Jeugd- zorg (CORV)	Omg. loket
LV BAG	GGK iWMO & iJW	WOZ	StUF- GEO

Binnengemeentelijke ketens

Prefill services	Betalen en Invorderen	Document creatie	BAG-GBA BAG - WOZ
Zaak- en Document services	RSGB Bevraging	WABO- BAG	

Eindproducten – compacte / scherpe standaarden



Halffabricaten en grondstoffen

StUF-onderlaag

StUF 3.xx berichtstandaard

Protocolbindingen



- Gereedschapskist voor ontwerp
 - halffabricaten, en/of
 - Eindproducten
- Flexibele constructies
- Uitbreidbaar
 - wel backwards compatible
- Generiek en domein-onafh.
- Onder regime “Comply-Explain”
 - StUF Best Practices
 - Producteisen-document
 - StUF beheermodel

Wat zit er in de StUF-gereedschapskist?



- Contentmodel
 - elementen, kern- en metagegevens
 - (super)entiteiten, relaties, groepen
 - materiële en formele historie
- Berichtsoorten
 - Kennisgevingen
 - Synchronisatieberichten
 - Vraag-antwoordberichten
 - Dienstberichten
- Logistiek
 - protocollen en bindingen

Flexibele StUF-constructies



- Contentmodel
 - extraElementen
 - aanvullendeElementen
- Dienstbericht (Di/Du)
 - <parameters>
 - <vrijeParameters>
 - StUF:functie attribute (lego-functionaliteit)
 - Antwoord
 - Entiteit
 - Selectie
 - Update

Uitbreidbare gereedschapskist



StUF 3.xx berichtstandaard

Protocolbindingen

- Nieuwe berichtsoorten
 - Streaming, request-response
- Nieuwe constructies
 - aanvullendeElementen
 - relaties met type "anyUri"
 - ...
- Nieuwe formaten
 - JSON, ...
- Protocolbindingen
 - AMQP, REST, ATOM, ...



woz0310



**prefill
e-form.**



bg0310

zkn0310



stuf0301



woz0310



**prefille
e-form.**



bg0310

zkn0310



stuf0301

aanvullendeElementen



woz0310



**prefill
e-form.**



bg0310

zkn0310



aanvullende Elementen

stuf0301

Nieuw paradigma: koppelvlakken moeten succesvol implementeerbaar zijn!



- Codegeneratie
 - Vermijden nillables
 - Vermijden van fixed-constructie
 - Vermijden van choices, unions
 - ...
- Hergebruik (inter)nationale constructies:
 - GAB-afspraken
 - GML
- Koppelvlakontwerper bepaalt het succes

Wanneer voldoet een koppelvlak aan StUF?

E03. Voldoet aan StUF-onderlaag

E07. Voldoet aan StUF Best Practices

E08. Optimaal hergebruik bestaande StUF-onderdelen

E09. Geen conflicten met andere StUF-onderdelen

Hoe zit het dan met CMIS in het ZS-DS koppelvlak?

Wat zijn de spelregels van StUF? (deel 1)

<ns:geefAdressenBijPostcode>

<ns:stuurgegevens>

<StUF:berichtcode>Di01</StUF:berichtcode>

<StUF:functie>geefAdressenBijPostcode</StUF:functie>

</ns:stuurgegevens>

<adres stuf:entiteittype="AOA">

<postcode>1058BK</postcode>

</adres>

</ns:geefAdressenBijPostcode>

Wat zijn de spelregels van StUF? (deel 1)

<ns:geefAdressenBijPostcode>

Stuurgegevens optioneel

<adres stuf:entiteittype="AOA">

<postcode>1058BK</postcode>

</adres>

</ns:geefAdressenBijPostcode>

Wat zijn de spelregels van StUF? (deel 1)

`<ns:geefAdressenBijPostcode>`

Stuurgegevens optioneel

Entiteit-container optioneel

`<postcode>1058BK</postcode>`

Entiteit-container optioneel

`</ns:geefAdressenBijPostcode>`

Wat zijn de spelregels van StUF? (deel 1)

Berichtelement optioneel

Stuurgegevens optioneel

Entiteit-container optioneel

<postcode>1058BK</postcode>

Entiteit-container optioneel

Berichtelement optioneel

Wat zijn de spelregels van StUF? (deel 2)

```
<ns:geefAdressenBijPostcodeResponse>
```

```
  <ns:stuurgegevens>
```

```
    <StUF:berichtcode>Du01</StUF:berichtcode>
```

```
    <StUF:functie>geefAdressenBijPostcode</StUF:functie>
```

```
  </ns:stuurgegevens>
```

```
  <adres StUF:entiteittype="AOA">
```

```
    <gor.straatnaam>Curacaostraat</gor.straatnaam>
```

```
    <huisnummer>23</huisnummer>
```

```
    <postcode>1058BK</postcode>
```

```
    <woonplaats>Amsterdam</woonplaats>
```

```
  </adres>
```

```
  <adres StUF:entiteittype="AOA">
```

```
    <gor.straatnaam>Curacaostraat</gor.straatnaam>
```

```
    <huisnummer>25</huisnummer>
```

```
    <postcode>1058BK</postcode>
```

```
    <woonplaats>Amsterdam</woonplaats>
```

```
  </adres>
```

```
  ...
```

```
</ns:geefAdressenBijPostcodeResponse>
```

Wat zijn de spelregels van StUF? (deel 2)

<ns:geefAdressenBijPostcodeResponse>

Stuurgegevens optioneel

```
<adres StUF:entiteittype="AOA">
  <gor.straatnaam>Curacaostraat</gor.straatnaam>
  <huisnummer>23</huisnummer>
  <postcode>1058BK</postcode>
  <woonplaats>Amsterdam</woonplaats>
</adres>
<adres StUF:entiteittype="AOA">
  <gor.straatnaam>Curacaostraat</gor.straatnaam>
  <huisnummer>25</huisnummer>
  <postcode>1058BK</postcode>
  <woonplaats>Amsterdam</woonplaats>
</adres>
...
</ns:geefAdressenBijPostcodeResponse>
```

Wat zijn de spelregels van StUF? (deel 2)

<adressen>

Stuurgegevens optioneel

```
<adres StUF:entiteittype="AOA">
  <gor.straatnaam>Curacaostraat</gor.straatnaam>
  <huisnummer>23</huisnummer>
  <postcode>1058BK</postcode>
  <woonplaats>Amsterdam</woonplaats>
</adres>
<adres StUF:entiteittype="AOA">
  <gor.straatnaam>Curacaostraat</gor.straatnaam>
  <huisnummer>25</huisnummer>
  <postcode>1058BK</postcode>
  <woonplaats>Amsterdam</woonplaats>
</adres>
...
```

</adressen>

Visievorming op StUF (deel 1)

- Flexibiliteit om nieuwe constructies en formaten toe te voegen in de onderlaag, scherpte in de eindproducten
- Creativiteit voor de berichtontwerper, eenduidigheid en gemak voor de software ontwikkelaar
- Comply-explain: gebruik zoveel mogelijk de standaard constructies uit de onderlaag
- Onderlaag is generiek: business- en domeinonafhankelijk

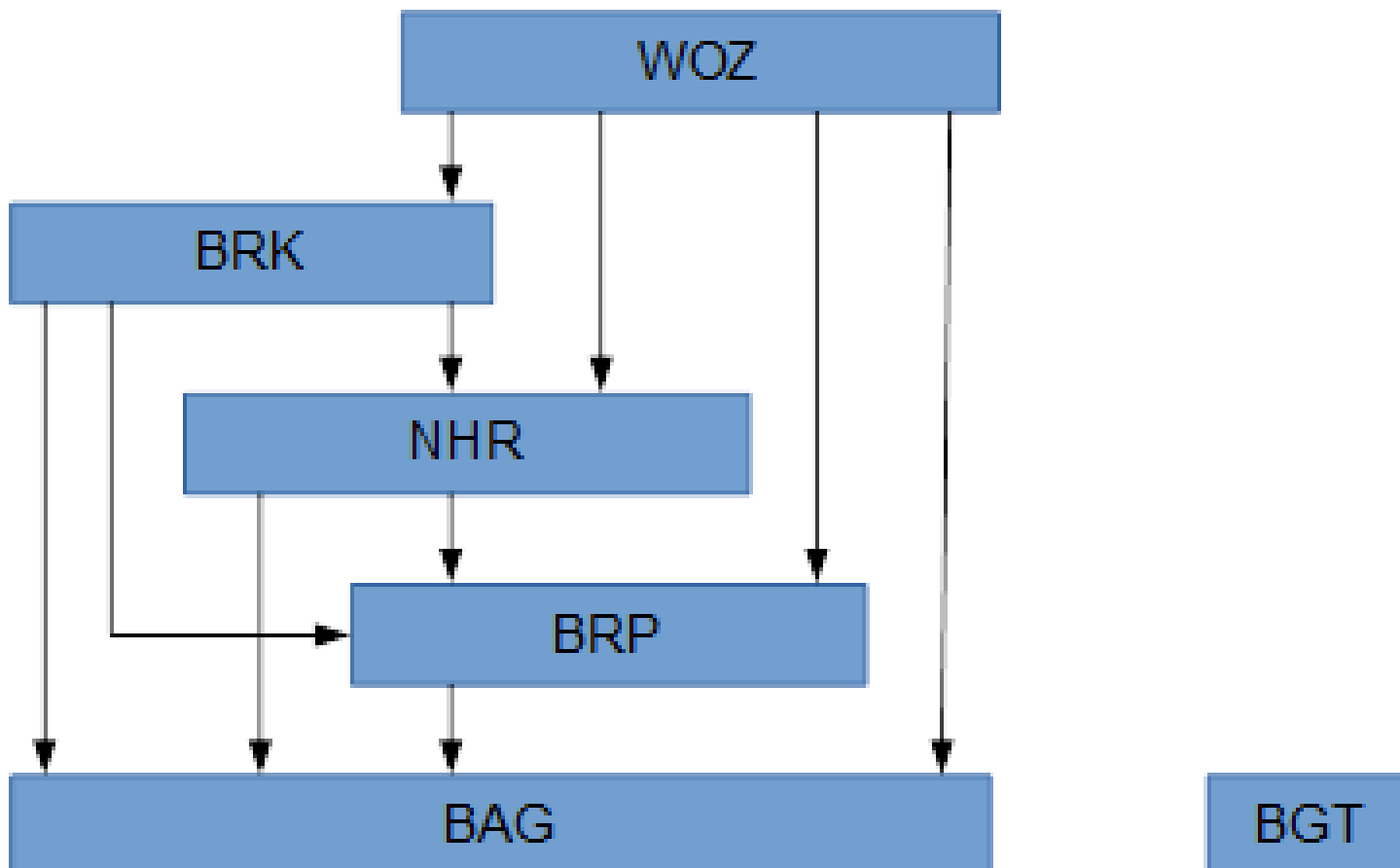
Visievorming op StUF (deel 2)

- Focus op eenvoudige implementatie koppelvlak
 - Codegeneratie
 - Scherpe intuïtieve koppervlakken (vrije constructies in onderlaag nodig)
 - Grondige prototyping door KING naast pilot Den Haag!!!!
 - Evaluatie door SIG of andere onafh. partij
- StUF-onderlaag uitbreidbaar maken voor toekomstige ontwikkelingen (JSON, REST)

Visievorming op StUF (deel 3)

- StUF-spelregels scherper beschrijven:
 - gebruik eerst standaard constructies (Lk, Lv-La)
 - flexibiliteit alleen indien nodig (Di-Du)
 - Gebruik in eerste instantie XML als serialisatie-formaat
- Indien meer flexibiliteit echt nodig is, die onder voorwaarden aanbieden (comply-explain) :
 - JSON formaat
 - REST-protocolbinding
 - Vrije request-response interacties
 - Losse gegevens zonder entiteit-container

Modularisering bg0320



bg0320 opsplitsen in 24 modules

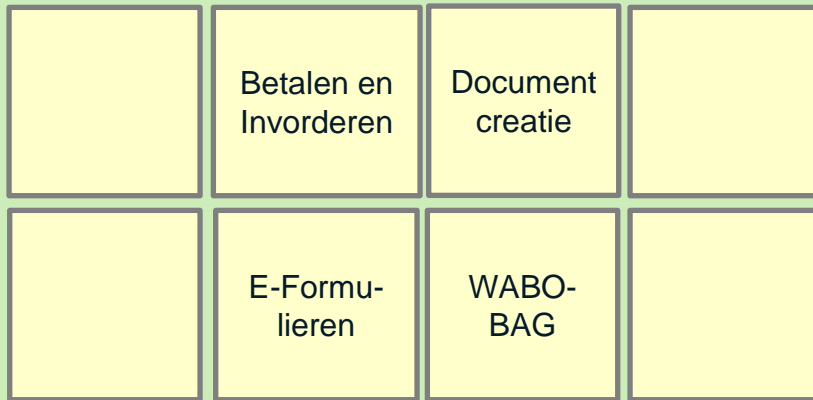
- Basisregistraties (6 combinaties)
 - WOZ, BRK, NHR, BRP, BAG, BGT
- Berichtsoort (2 combinaties)
 - mutaties (kennisgevingen en synchronisatie)
 - vraagAntwoord
- Historie (2 combinaties)
 - Zonder historie
 - Materiële of formele historie

Hoe positioneren we de 24 modules?

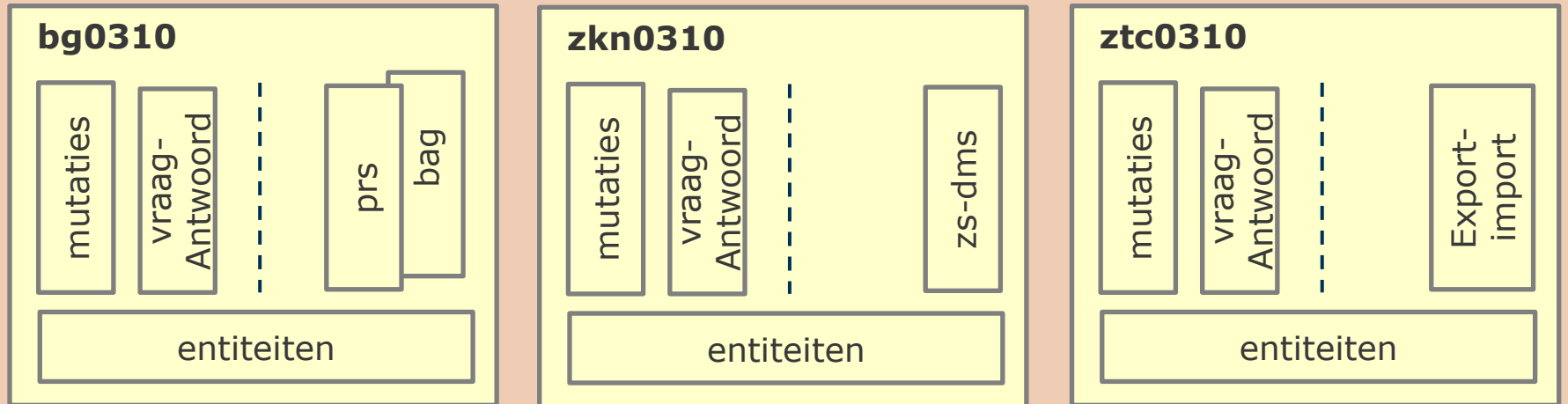
- Als koppelvlakken met eigen namespaces
- Bij elke module de minimaal verplichte set specificeren
- ...

StUF-familie (huidige situatie)

KING koppelvlakken



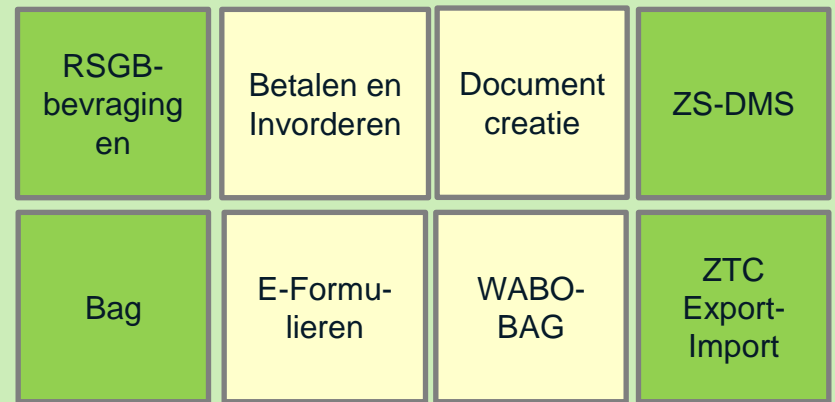
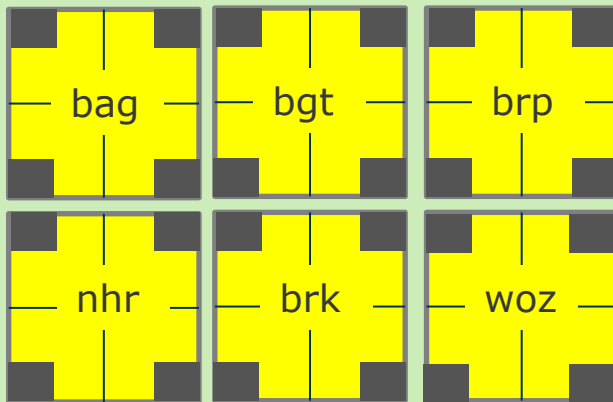
Eindproducten – compacte / scherpe standaarden



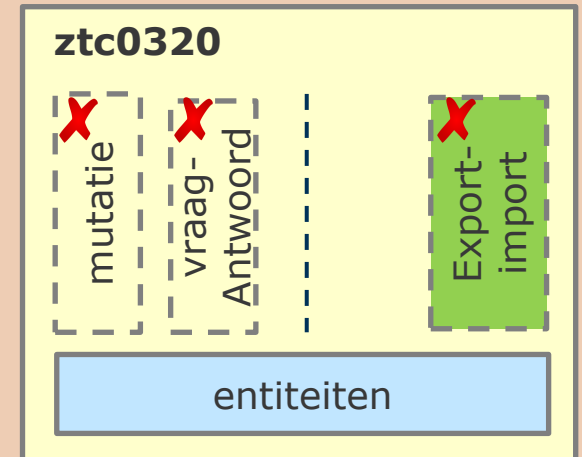
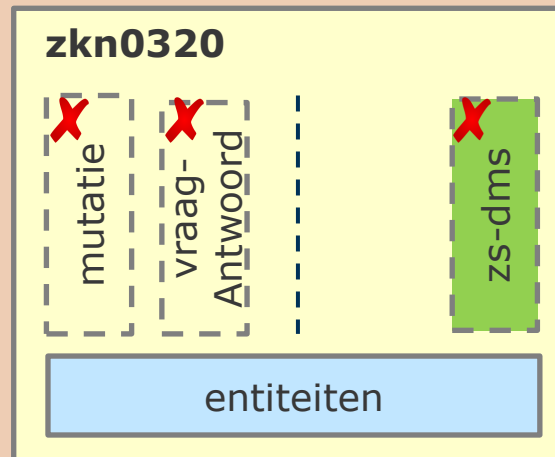
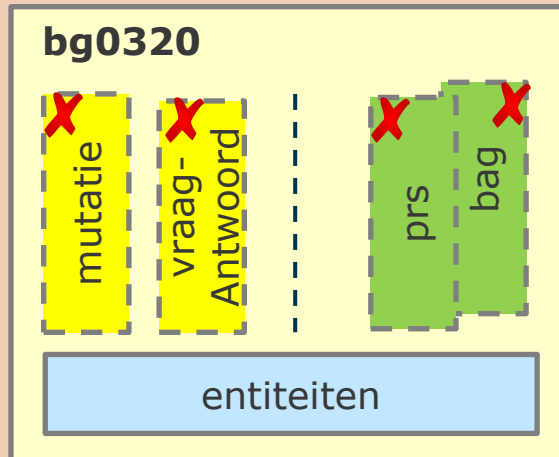
Halffabricaten

StUF-familie (nieuwe situatie)

KING koppelvlakken



Eindproducten – compacte / scherpe standaarden



Echte halffabricaten